INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE & ENGINEERING
IJASCSE VOLUME 9 ISSUE 5, 2020
05/31/2020

Comparative Analysis of SCAN And CSCAN Disk Scheduling Algorithm

Gregorius Neven Yusuf[1], Wirawan Istiono[2]
Universitas Multimedia Nusantara
Jl. Scientia Boulevard, Gading Serpong
Tangerang, Indonesia
neven.yusuf@student.umn.ac.id[1], wirawan.istiono@umn.ac.id[2]

***Abstract*—In operating systems, seek time is very crucial part, because all device requests are linked in queues. Disk Scheduling Algorithms are used to reduce the total seek time of any request to make performance of operating system is faster. A comparison between two disc scheduling algorithm, such as SCAN (Elevator) algorithm and CSCAN (Circular SCAN) algorithm, is tested by exploring the unique character of each algorithm and then it will be decided which algorithm is the best in performing disc scheduling which can be implemented in the operating system. To compare these algorithms, a software named Track Animation will be used. With track animation software we can see a simulation of various disc scheduling algorithms. The result from this research between SCAN and CSCAN shown the CSCAN algorithm produces less seek time compared to the SCAN algorithm. In addition, CSCAN results in more uniform waiting times compared to SCAN, but with difference dataset which have small standard deviations and variance, SCAN algorithm will make disk performance faster and efficient than CSCAN algorithm.**

Keywords-component; Disc Scheduling, SCAN elevator, Circular-SCAN, C-SCAN, comparison, track animation

I. Introduction

As time flies and a rapid development of technology in this era, the speed and ease of a job of all things is becoming a universal benchmark of how things are liked and not liked by people. The increasingly complex human jobs nowadays, demand a dynamic technology to follow the rhythm of human works and the pace of the development. This demand is answered by the Moore's Law, an observation that states the number of transistors on a computer chip will double every two years [1]. This allows the computer processor to have a greater speed over time, but on a computer system, the processor does not run alone. The main thing that is needed so that the computer can load data is memory, a home to all data [2].

The problem that arises is the speed of memory development that can not be kept up with the development of the processor. According to Moore's Law, each year, the gap between processor and memory speeds will grow by 50 percent. The memory that is being discussed is a type of main memory that is used to store instructions while the program is running [3][4]. In this journal, what will be discussed further is disc scheduling in secondary memory, which has a much slower speed compared to main memory. Secondary memory, which is a multi-function storage media, is still a vital component of a computer. However, secondary memory can always be called a bottleneck that slows down the work of the whole computer. Therefore, various studies have been conducted to find techniques to increase the speed of secondary memory [5].

In secondary memory, that uses a moveable head to read and write data such as hard disks, a problem that arises is the time wasted, when the head id moving and seeking for data or seek time. Seek time must be minimized so that access time is also reduced and data can be processed immediately by the processor. Disk scheduling is used to minimize this seek time and access time so that the operating system can effectively abstract hardware to the computer [6].

Therefore, this journal aim is arranged to find the unique characteristics of the two disk scheduling algorithms between SCAN and CSCAN, so that we can know how and when they could be used so that the operating system can be faster and efficiently run programs on a computer.

II. DISC SCHEDULING ALGORITHM INDICATOR

Disc Scheduling is a technique that is used by operating systems to handle data flow that is needed by the processor from a secondary storage, such as a hard disk. When a certain program is run by the computer, disc scheduling will be run, so the data will flow and be used faster, therefore programs can be executed faster. In a disc system, there are four basic parameters that are used as an indicators of disc scheduling performance [7].

a) Seek Time: Time taken by the head disc to move towards the cylinder or the intended part.

$$T(seek) = m * n + s \quad (1)$$

m: running time between cylinders (1-3 ms)
n: number of cylinders
s: acceleration and deceleration time (3-20 ms)

b) Rotational Delay (Latency): The time needed for the disc to rotate so the disc head can find the intended cylinder.

$$T\ (\text{latency}) = 1/2 * 1 / r \quad (2)$$

r : rotation time (rotation / sec)

c) Transfer Time: The time required for data to be transmitted in or out of a disc by the disc head.

$$T\ (\text{transfer}) = b / (r * N) \quad (3)$$

b: number of data bytes
r: rotation time
N: number of bytes per cylinder

d) Access Time: The total time needed to access (read or write) the intended data

$$T\ (\text{seek}) + T\ (\text{latency}) + T\ (\text{Transfer}) \quad (4)$$

Data from these basic parameters will be processed to become information that can be more easily understood.

The main purpose of disc scheduling is the consistency of the time needed to complete a data request from the CPU. A good disc scheduling technique is one that can produce waiting times that do not vary too much [8][9]. Therefore, besides the four basic indicators, there are two additional indicators that can summarized the information obtained from the basic indicators, namely "mean response time", the average time taken to complete a data request, and "throughput", The number of requests completed for each time .

Disc scheduling must be able to maximize "throughput" and minimize the "mean response time". Besides that the variation of mean response time also influences the performance of disc scheduling performance [10].

III. VARIANCE AND STANDARD DEVIATION

Variance and standard deviation are values in statistics that can measure how scattered data is. If the variance and standard deviation of a large set of data is high, then the data is highly scattered [11].

Variance:

$$s^2 = \frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_1\right)^2}{n(n-1)} \quad (5)$$

Standard Deviation:

$$s = \sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_1\right)^2}{n(n-1)}} \quad (6)$$

Xi = index of data
X1 = mean
n = number of samples

Variance and standard deviation are not much apart, but still have main differences. The value of variance determines how spread each individual data is, while the standard deviation decides more from the overall scattering of samples with the average of total samples.

The value of variance and standard deviation will be used to see the characteristics possessed by the data set in the form of the location of a data on disk. The location of the data is represented by integers from 1 to the number of tracks contained on the disk [12]. This path or tracks data is the output of the operating system that is carrying out its jobs to read and write data. The sequence of data paths will also vary, due to the large amount of data and also large storage locations, and also determines how the disk scheduling procedure is performed.

IV. DISC SCHEDULING ALGORITHM

SCAN algorithm, or commonly called Elevator is a disk scheduling algorithm that is characterized by moving the head on the disk with rules such as an elevator, which only moves in one direction and serves the request of the appropriate track number until it reaches its end [13]. Then, the disk head will move in the opposite direction while serving requests for tracks numbers that have not yet been served, for example, A disk has 250 tracks numbered from 0 to 249. This disk has data to be retrieved by a computer processor and the order of data requests locations are as follows:

202, 168, 245, 55, 143, 173, 146, 23, 76, 140

The disk head is in path 80 at first. Based on the results obtained, the sequence of track numbers served is as follows:

76, 55, 23, 140, 143, 146, 168, 173, 202, 245

Initially, the Disk head is on the track number 80, therefore the request for track number 76 will be served first because it is closest to the initial position of the head. The disk head moves down because it moves from 80 to 76, then the head will move down to the end of the disk. After arriving at the track number 23 (the edge of the request) the head will still move until the end of the disk that is numbered 0 although no paths are to be served.

The disk head will move in the opposite direction after arriving at track 0, then the head moves towards track 140 because it is the closest path to be served. The disk head will continue to move until the last track number to be serviced is track number 245.

The C-SCAN algorithm, short for Circular SCAN, is another form of the SCAN algorithm. In this disk scheduling algorithm, the disk head is only allowed to move and serve data location requests in one direction. Then, after reaching the end of the disk, the disk head will move in the opposite direction towards the other end of the disk, without serving data location requests. After reaching the other end of the disk, the head will move back in the same direction as the starting direction, and serve requests for data locations that have not been served. Based on the results obtained, the sequence of track numbers served is as follows:

80, 76, 55, 23, 0, 249, 245, 202, 173, 168, 146, 143, 140

Initially, the disk head is on the track number 80, therefore requests number 76 will be served first because it is the closest to the initial position of the head. Disk head will move downwards because it moves from 80 to 76, then the next head will move down to the end of the disk. After arriving at the track number 23 (the edge of the request) the head will still move until the end of the disk that is the track numbered 0 although no request will be served [14].

Unlike the SCAN algorithm, the disk head will still move in the opposite direction, but it will not serve requests for the location of data on the disk, instead the head will continue to move until the other end of the disk without stopping. After reaching the end of the disk, then the head will move down again to serve requests for data locations that have not been served starting from the track numbered 245, because it is the closest to the end of the disk (249), after that it will be continued until the last data request, path numbered 140.

V. ANALYSIS

As an experimental data to be carried out, four data sets have been prepared (each totaling ten data requests) in the form of a sequence of data location requests on disk by the processor. The six data sets are data with variance values and standard deviations that are different and well adjusted, so that the characteristics of each algorithm for each data set can be seen easily and clearly.

Given a disk that has a path as many as 1024 numbered from 0 to 1024. The location of the head disk is initially at the path numbered 0 (the initial value is taken to simulate the calling of disk location data by the processor when the disk is turned on). The seek time required by the disk to travel from one track to another is one millisecond.

Data set1; the sequence of processors request are as follows: 48, 117, 985, 919, 659, 730, 9, 874, 897, 528. Where standard Deviation is 349.31 and variance is 122017.12.

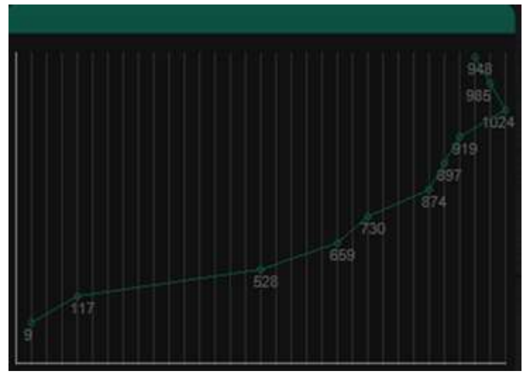

Figure 1. Result Track Animaction SCAN with Data set1

Based on the result that shown in Figure 1, the sequence of the service of the requests is as follows: 948, 985, 1024, 919, 897, 874, 730, 659, 528, 117, 9 where total seek time is 1091 ms.

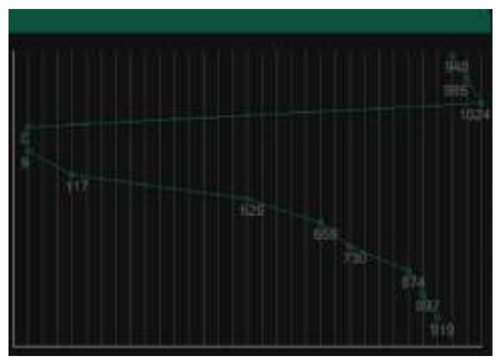

Figure 2. Result Track Animaction CSCAN with Data set1

Based on the result that shown in Figure 2, the sequence of the service of the requests is as follows: 948, 985, 1024, 0, 9, 117, 528, 659, 730, 874, 897, 919, where total seek time is 995 ms

Data set2; the sequence of processors request are as follows: 575, 816, 490, 781, 566, 148, 618, 89, 262, 74. Where standard deviation is 278.89 and variance 77778.99

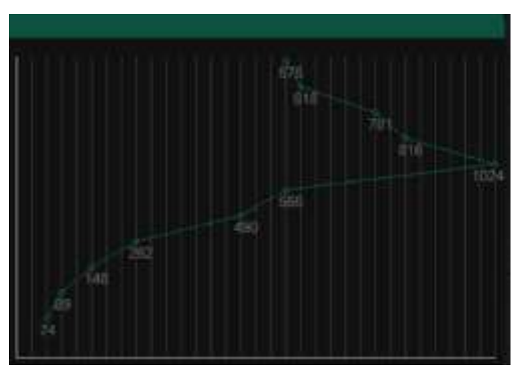

Figure 3. Result Track Animaction SCAN with Data set2

Based on the result that shown in Figure 3, the sequence of the service of the requests is as follows: 575, 618, 781, 816, 1024, 566, 490, 262, 148, 89, 74 where total seek time = 1399 ms.

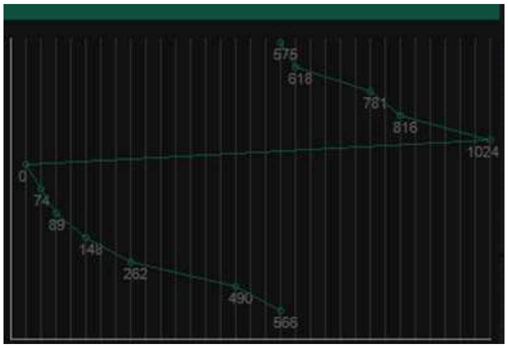

Figure 4. Result Track Animaction CSCAN with Data set2

Based on the result that shown in Figure 4, the sequence of the service of the requests is as follows: 575, 618, 781, 816, 1024, 0, 74, 89, 148, 262, 490, 566 where total seek time = 1015 ms.

Data set3; The sequence of processors request are as follows: 976, 678, 873, 581, 565, 667, 916, 887, 865, 886 where standard deviation is 150.40 and Variance is 22620.71.

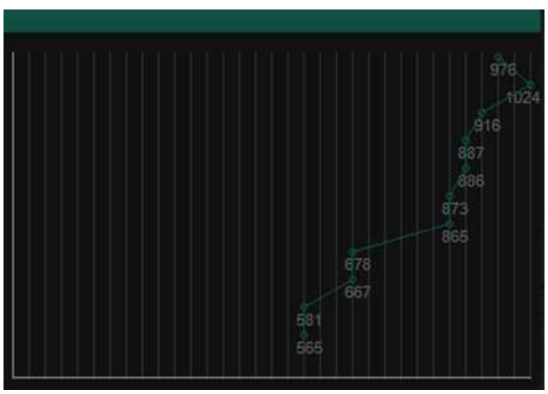

Figure 5. Result Track Animaction SCAN with Data set3

Based on the result that shown in Figure 5, the sequence of the service of the requests is as follows: 976, 1024, 916, 887, 886, 873, 865, 678, 667, 581, 565 where total seek time = 507 ms.

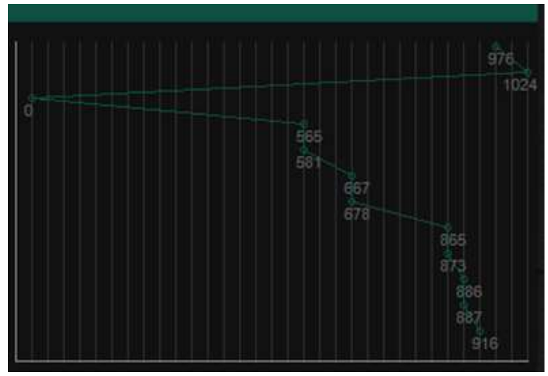

Figure 6. Result Track Animaction SCAN with Data set3

Based on the result that shown in Figure 6, the sequence of the service of the requests is as follows: 976, 1024, 0, 565, 581, 667, 678, 865, 873, 886, 887, 916 where total seek time is 964 ms,

Data set4; the sequence of processors request are as follows: 54, 221, 331, 176, 120, 111, 130, 90, 85, 54 where Standard Deviation is 85.58 and variance is 7324.18.

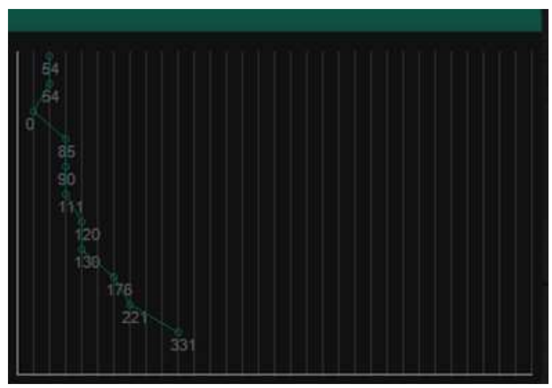

Figure 7. Result Track Animaction SCAN with Data set4

Based on the result that shown in Figure 7, the sequence of the service of the requests is as follows: 54, 54, 0, 85, 90, 111, 120, 130, 176, 221, 331 where total seek time is 385 ms

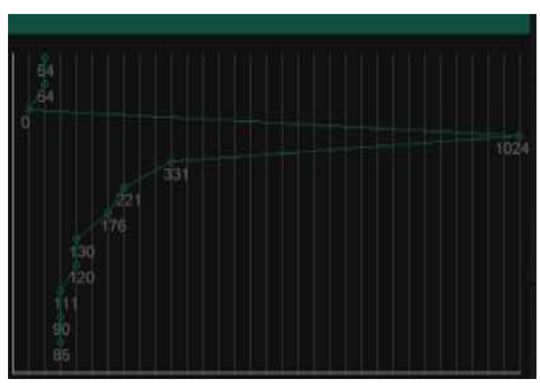

Figure 8. Result Track Animaction SCAN with Data set4

Based on the result that shown in Figure 8, the sequence of the service of the requests is as follows: 54, 54, 0, 1024, 331, 221, 176, 130, 120, 111, 90, 85 where total seek time is 993 ms.

VI. CONCLUSION

Based on analyst from dataset that can be concluded, overall the CSCAN algorithm produces less seek time compared to the SCAN algorithm. In addition, CSCAN results in more uniform waiting times compared to SCAN. The unique disk head movement of the CSCAN algorithm allows the disk to save time reading and writing to locations of data that are far apart or quite scattered.

However, this does not apply to data sets5 and data set6 which have small standard deviations and variance. As a result of the processor requests that are not scattered (small standard deviation values), the nature of CSCAN which requires the head disk to move to the end and back again to move in the same direction means wasting of significant time.

The use of SCAN algorithm on data sets that have a standard deviation below 200 will make disk performance faster and efficient, but the use of CSCAN will break performance. In practice, processor's location data requests will vary, so there is no disk scheduling algorithm that will produce excellent performance forever. Each algorithm has unique characteristics, depending on the processor request data.

VII. FUTURE WORK

Widening or reducing the number of datasets that will be tested between SCAN and CSCAN disk scheduling by using other track animation software, or by using manual graph depiction from the results of calculations with standard deviations.

REFERENCES

[1] E. Mollick, "Establishing Moore's Law," *IEEE Annals of the History of Computing*, vol. 28, no. 3, pp. 62–75, 2006.

[2] J. Celis, D. Gonzales, E. Lagda, and L. R. Jr, "A Comprehensive Review for Disk Scheduling Algorithms," *International Journal of Computer Science Issues*, vol. 11, no. 1, pp. 74–79, 2014.

[3] M. Eckert, D. Meyer, J. Haase, and B. Klauer, "Operating System Concepts for Reconfigurable Computing: Review and Survey," *International Journal of Reconfigurable Computing*, vol. 2016, p. 11, 2016.

[4] F. N. Ferdinand and F. V. Ferdinand, "A study on network design for the shortest path in expedition company," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1, pp. 1–4, 2018.

[5] R. Agrawal and C. Nyamful, "Challenges of big data storage and management," *Global Journal of Information Technology*, vol. 6, no. 1, 2016.

[6] Ç. Tuncay, *Modern Operating System Fourth Edition*, vol. 19, no. 3. Library of Congress Cataloging-in-Publication Data, 2008.

[7] A. T. Anshita Malviya, "Study of different algorithms use for disk scheduling process," *International Journal of Engineering Sciences & Research Technology*, vol. 5, no. 4, pp. 228–231, 2016.

[8] S. Shastri, A. Sharma, and V. Mansotra, "a Comparative Analysis of Disk Scheduling Algorithms," *Journal*, vol. 2, no. 4, pp. 16–25, 2016.

[9] D. D. Sinaga and S. Hansun, "Indonesian text document similarity detection system using rabin-karp and confix-stripping algorithms," *International Journal of Innovative Computing, Information and Control*, vol. 14, no. 5, pp. 1893–1903, 2018.

[10] T. J. Teorey and T. B. Pinkerton, "A Comparative Analysis of Disk Scheduling Policies," *Communications of the ACM*, vol. 15, no. 3, pp. 177–184, 1972.

[11] P. Barde and M. Barde, "What to use to express the variability of data: Standard deviation or standard error of mean?," *Perspectives in Clinical Research*, vol. 3, no. 3, p. 113, 2012.

[12] Herdiana, W. Stallings, and ب. ق. زاده, *Multiprocessor and Real-Time Scheduling*, vol. 53, no. 9. 2012.

[13] M. C. Waveguides, N. Osman, M. Koshiba, S. Member, and R. Kdji, "A COMPREHENSIVE ANALYSIS OF DISK SCHEDULING ALGORITHMS," *International Journal of Latest Trends in Engineering and Technology*, vol. 11, no. 1, 2018.

[14] S. Suranauwarat, "A disk scheduling algorithm simulator," *Computers in Education Journal*, vol. 8, no. 3, pp. 1–9, 2017.